Good Performance for Bad Days

or: are we measuring the right thing?

Marc Brooker

VP/Distinguished Engineer
Amazon Web Services

Who Am I?

- 16 years at AWS
  - 16 years of carrying a pager
- Worked on some of the largest cloud systems
  - EC2, EBS, Lambda, Aurora, Aurora DSQL, Bedrock, etc.
- Interested in system performance and stability
- Read about 4000 post-mortems
- https://brooker.co.za/blog/

Hypothesis 1:

In large-scale systems, system performance is the single largest contributor to system availability.

Hypothesis 1b:

In large-scale systems, unpredictable system performance is the single largest contributor to system unavailability.

Hypothesis 2:

Performance evaluation’s primary value is helping system designers make good decisions.

Hypothesis 3:

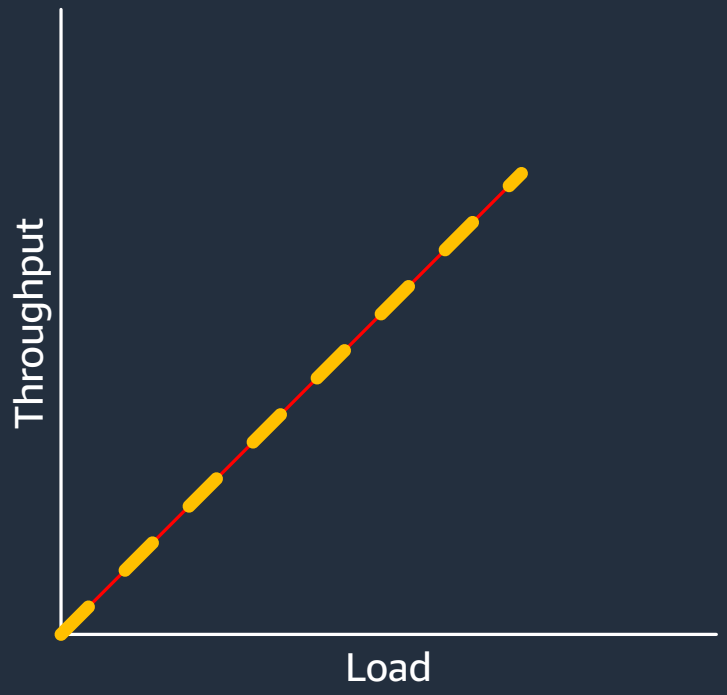
We’re not doing a great job of that.

Why?

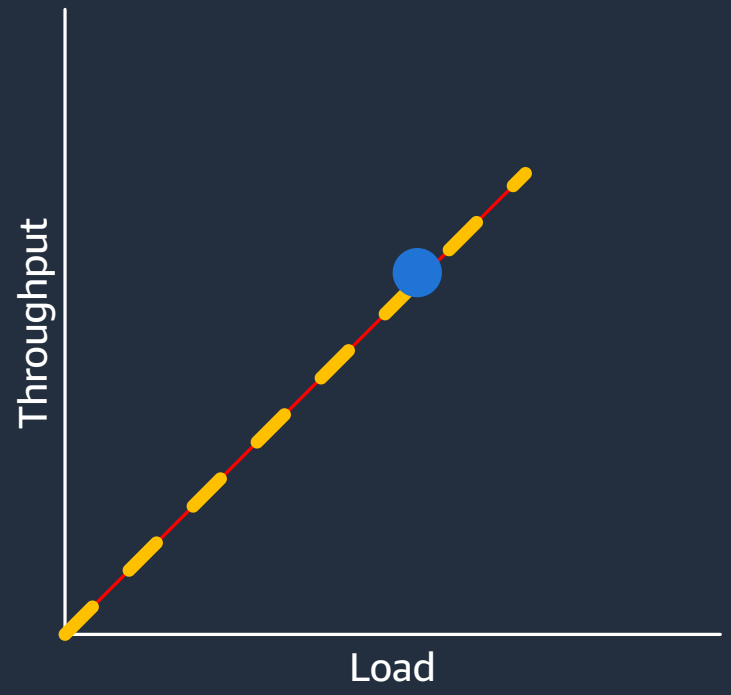
Why 1: Insufficient Attention to Overload Behavior

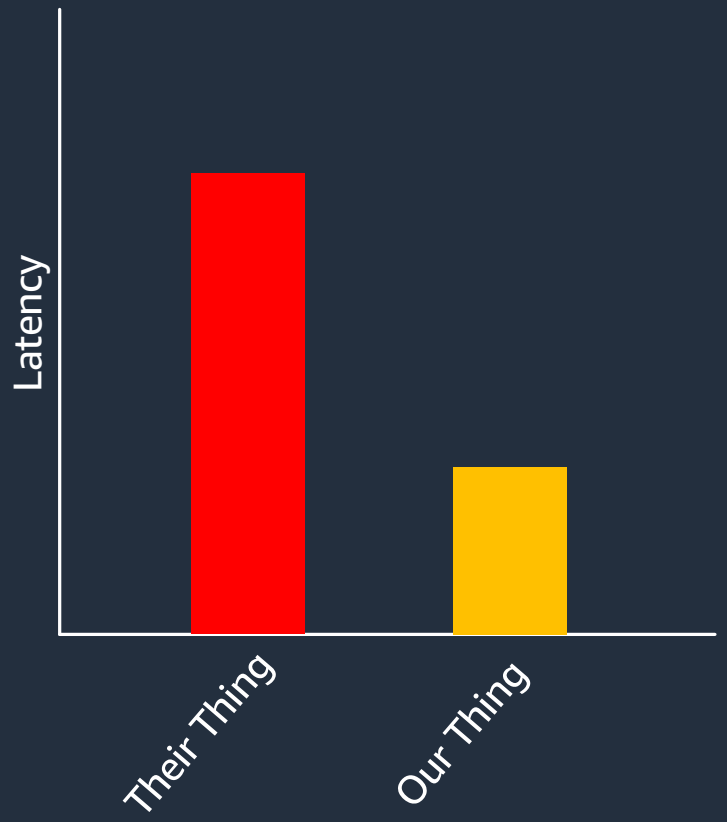
Not paying enough attention to the behavior of systems under excess or unexpected load.

GOOD PERFORMANCE FOR BAD DAYS

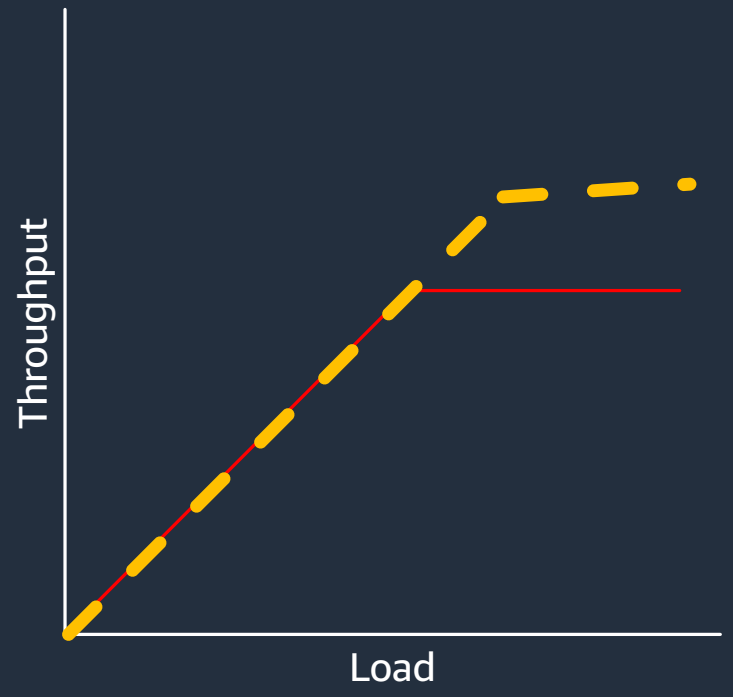


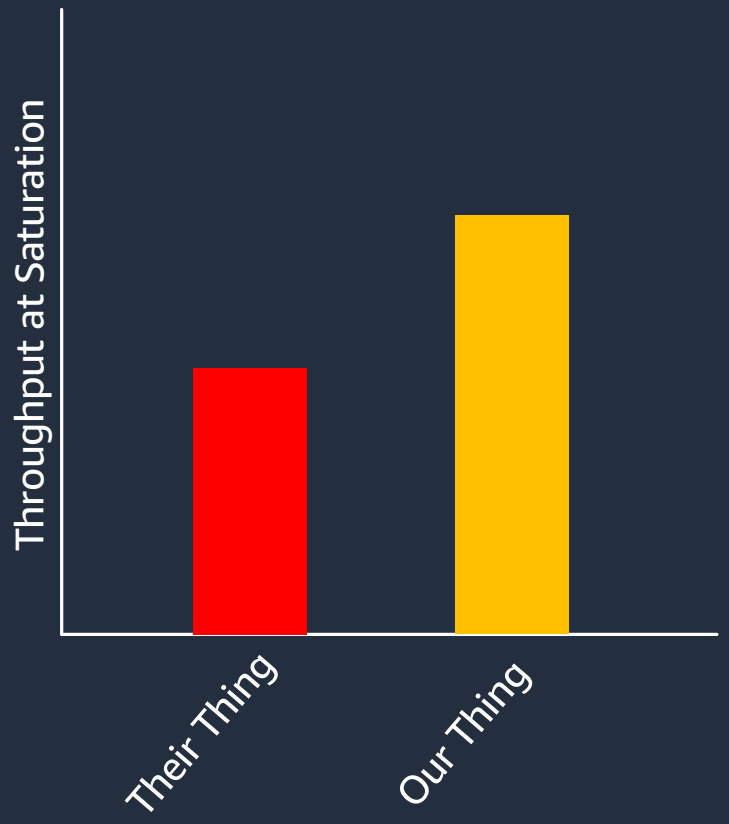
GOOD PERFORMANCE FOR BAD DAYS



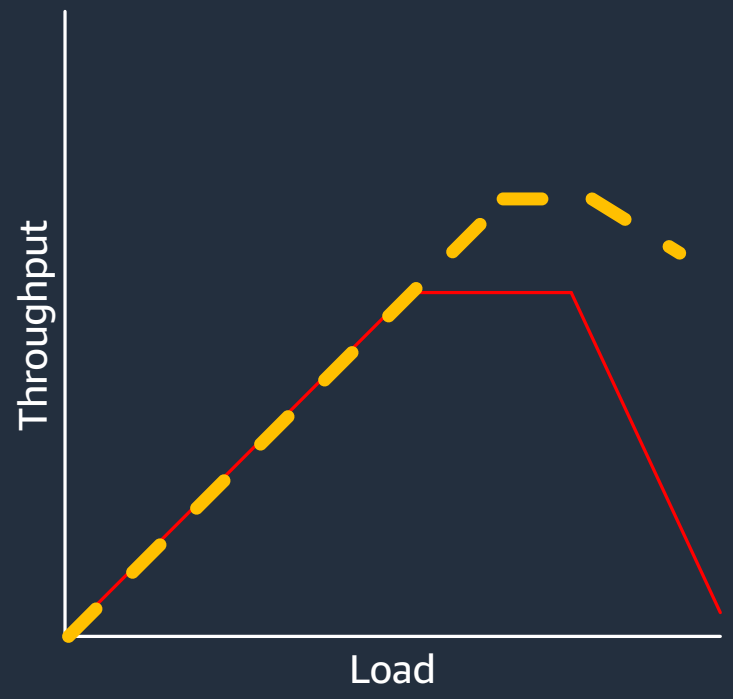


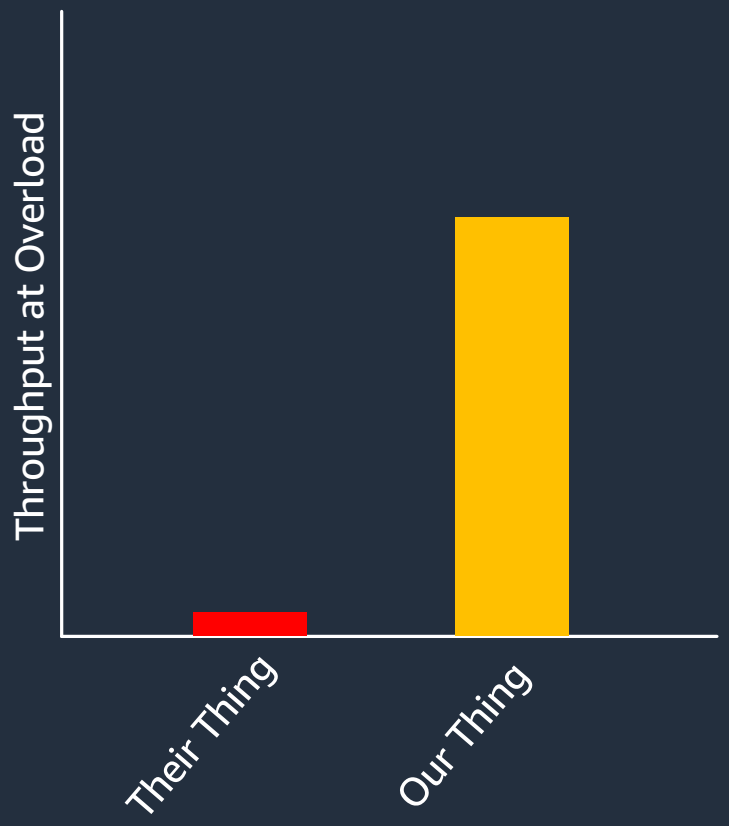
GOOD PERFORMANCE FOR BAD DAYS





GOOD PERFORMANCE FOR BAD DAYS





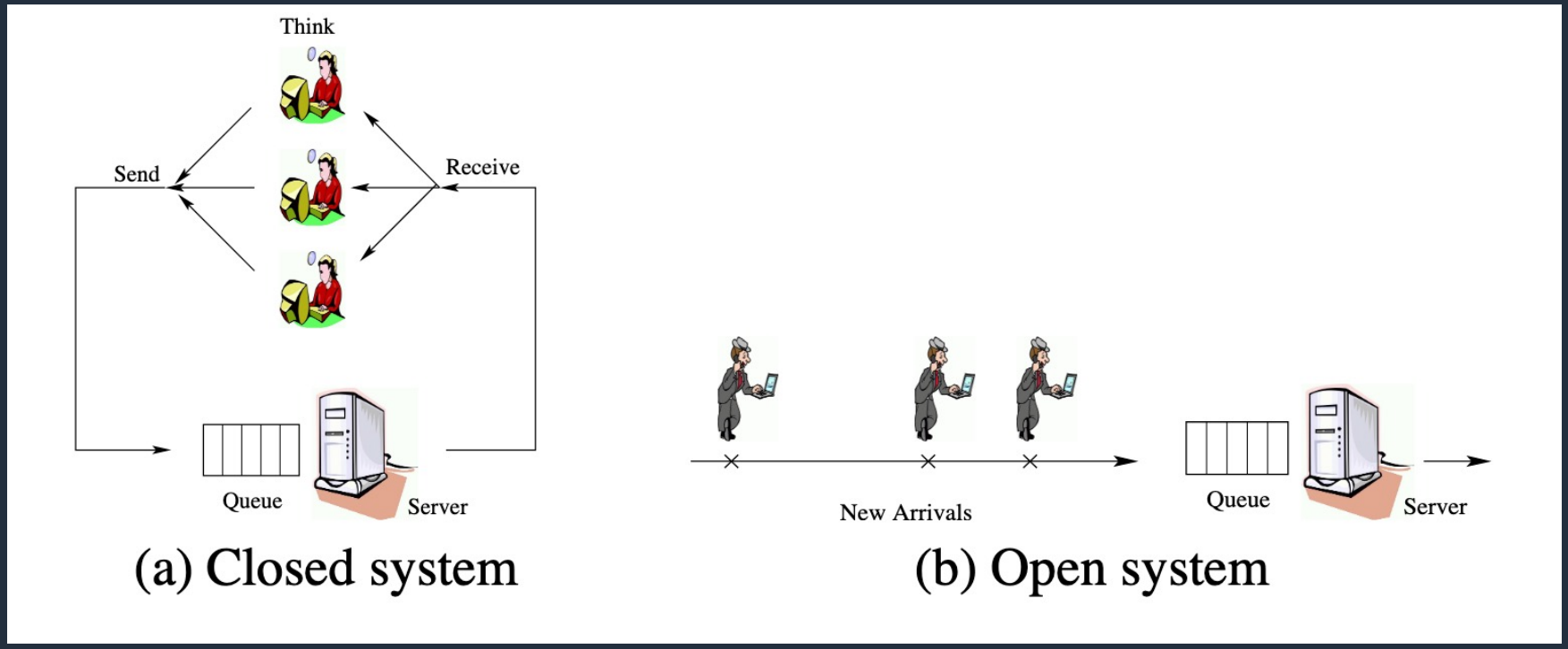
Research Roadmap 1:

- How do systems behave above saturation?
- How do those results compose?
- How do layers above know saturation is coming?

Why 2: Not Paying Attention to *Open* Workloads

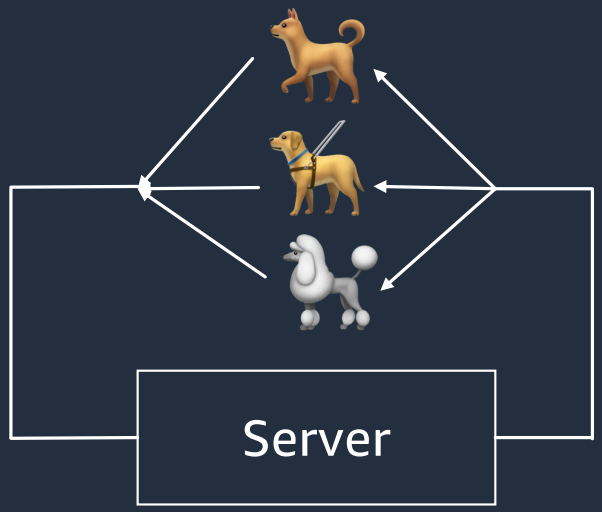
Many benchmarks set a number of threads, then spin in a closed loop.

These benchmarks suck.



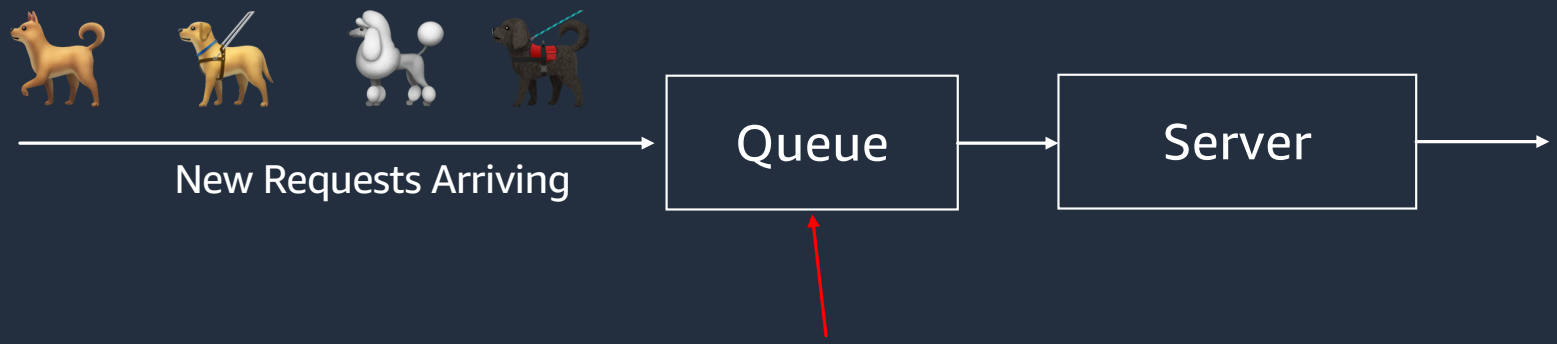
From “Open Versus Closed: A Cautionary Tale”, Schroeder et al, NSDI’06

They underestimate the effects of latency spikes.
(This is the famous *coordinated omission* problem).



Only one request per thread observes a stall.

They’re unrealistically kind.
When latency goes up, load goes down!



Slowed down? Hope you can deal with more work later!

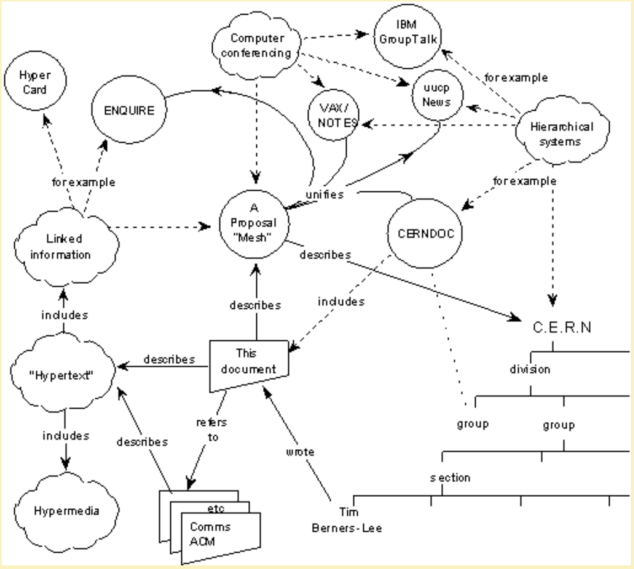
From “Open Versus Closed: A Cautionary Tale”, Schroeder et al, NSDI’06

They don’t reflect the reality most system builders live in.

Information Management: A Proposal

Tim Berners-Lee, CERN
March 1989, May 1990

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.



Overview

Many of the discussions of the future at CERN and the LHC era end with the question - ªYes, but how will we ever keep track of such a large project?º This proposal provides an answer to such questions. Firstly, it discusses the problem of information access at CERN. Then, it introduces the idea of linked information systems, and compares them with less flexible ways of finding information.

It then summarises my short experience with non-linear text systems known as ªhypertextº, describes what CERN needs from such a system, and what industry may provide. Finally, it suggests steps we should take to involve ourselves with hypertext now, so that individually and collectively we may understand what we are creating.

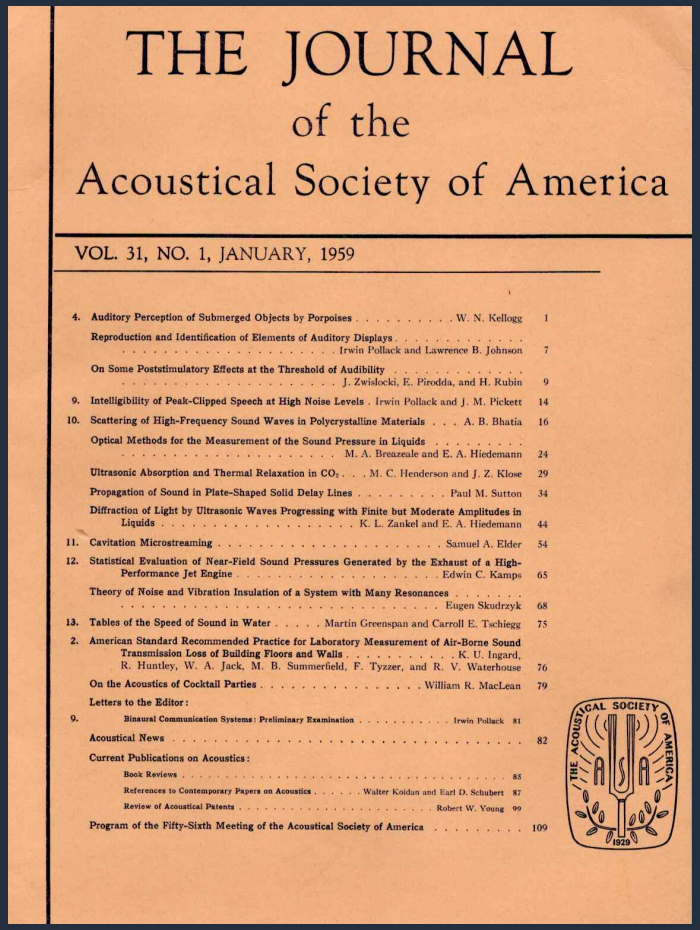
Research Roadmap 2:

- Benchmarks that represent 21st century workloads
- How does performance compose?
- How do clients and customers react to errors and latency?

Why 3: Not Paying Attention to *Metastability*

Overload conditions are *sticky*.

GOOD PERFORMANCE FOR BAD DAYS



THE JOURNAL
of the
Acoustical Society of America

VOL. 31, NO. 1, JANUARY, 1959

Network Working Group — John Nagle
Request For Comments: 896 — 6 January 1984
Ford Aerospace and Communications Corporation

Congestion Control in IP/TCP Internetworks

This memo discusses some aspects of congestion control in IP/TCP Internetworks. It is intended to stimulate thought and further discussion of this topic. While some specific suggestions are made for improved congestion control implementation, this memo does not specify any standards.

"As long as there is only one copy of each datagram in transit, congestion is under control. Once retransmission of datagrams … begins, there is potential for serious trouble.

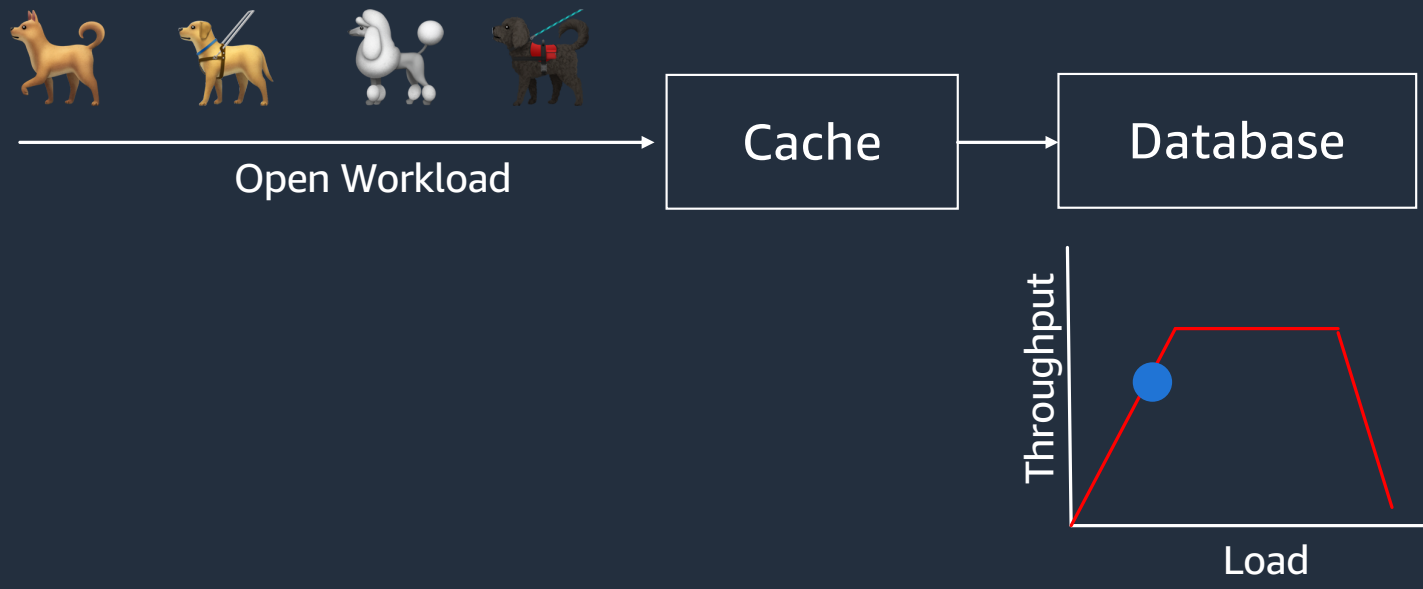
….

This condition is stable. Once the saturation point has been reached … the network will continue to operate in a degraded condition."

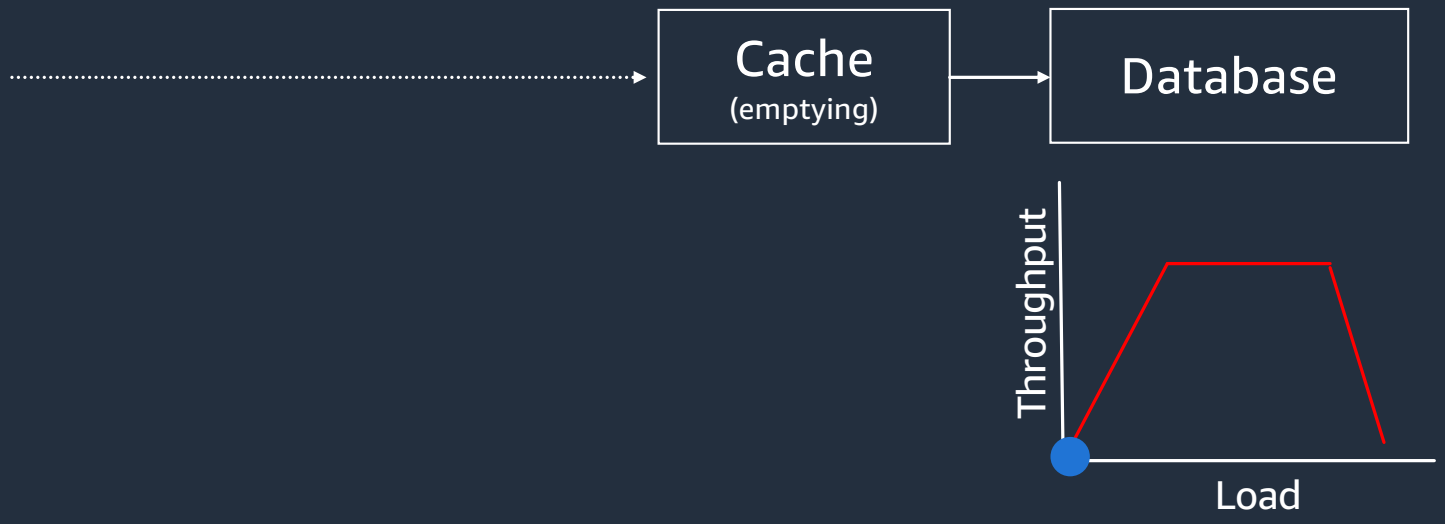
John Nagle, *RFC896: Congestion Control in IP/TCP Internetworks,* January 1984

GOOD PERFORMANCE FOR BAD DAYS

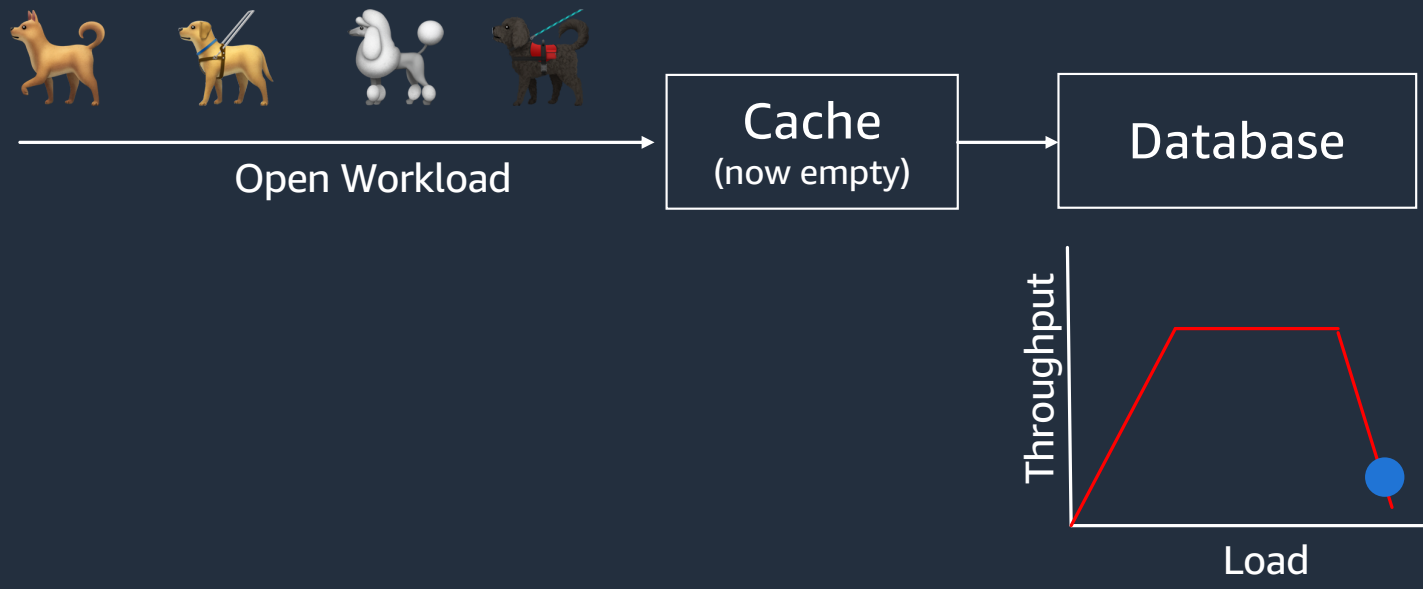
GOOD PERFORMANCE FOR BAD DAYS

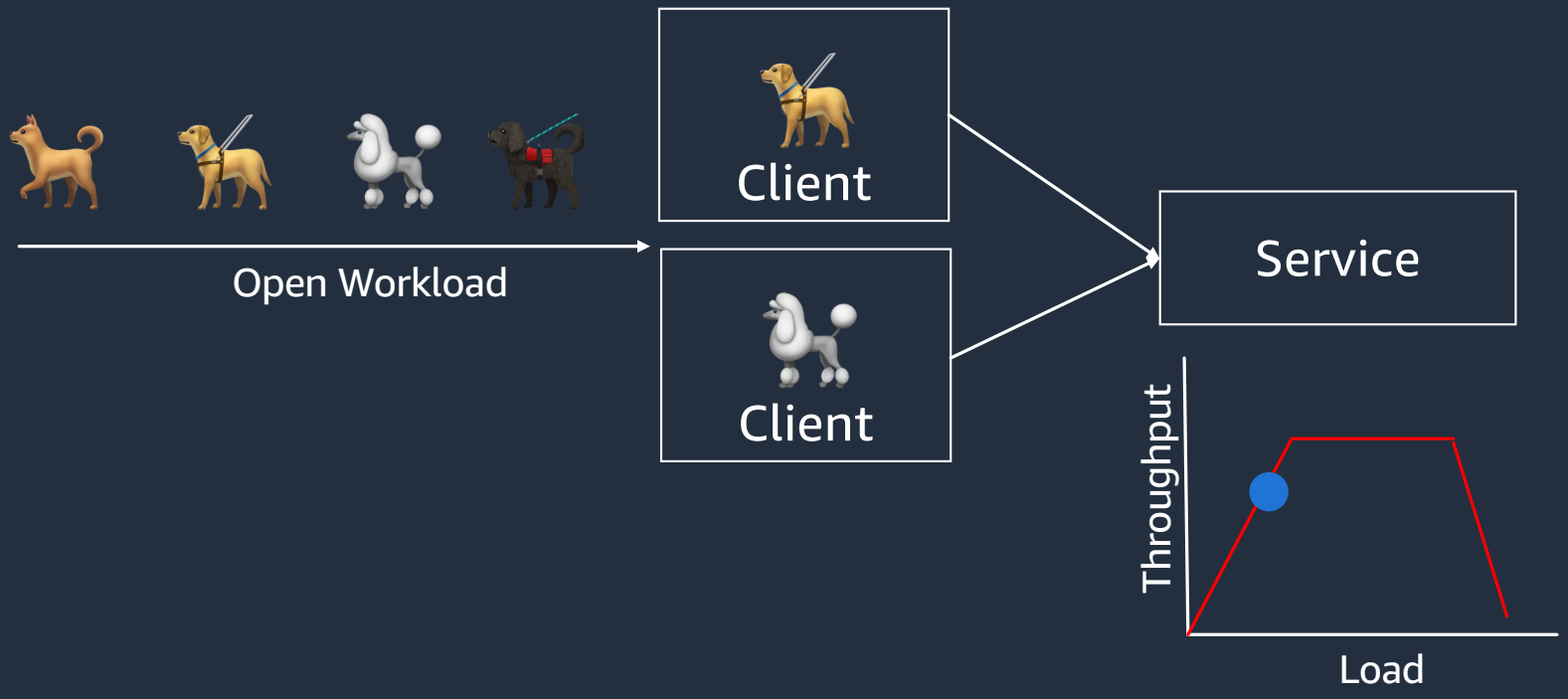


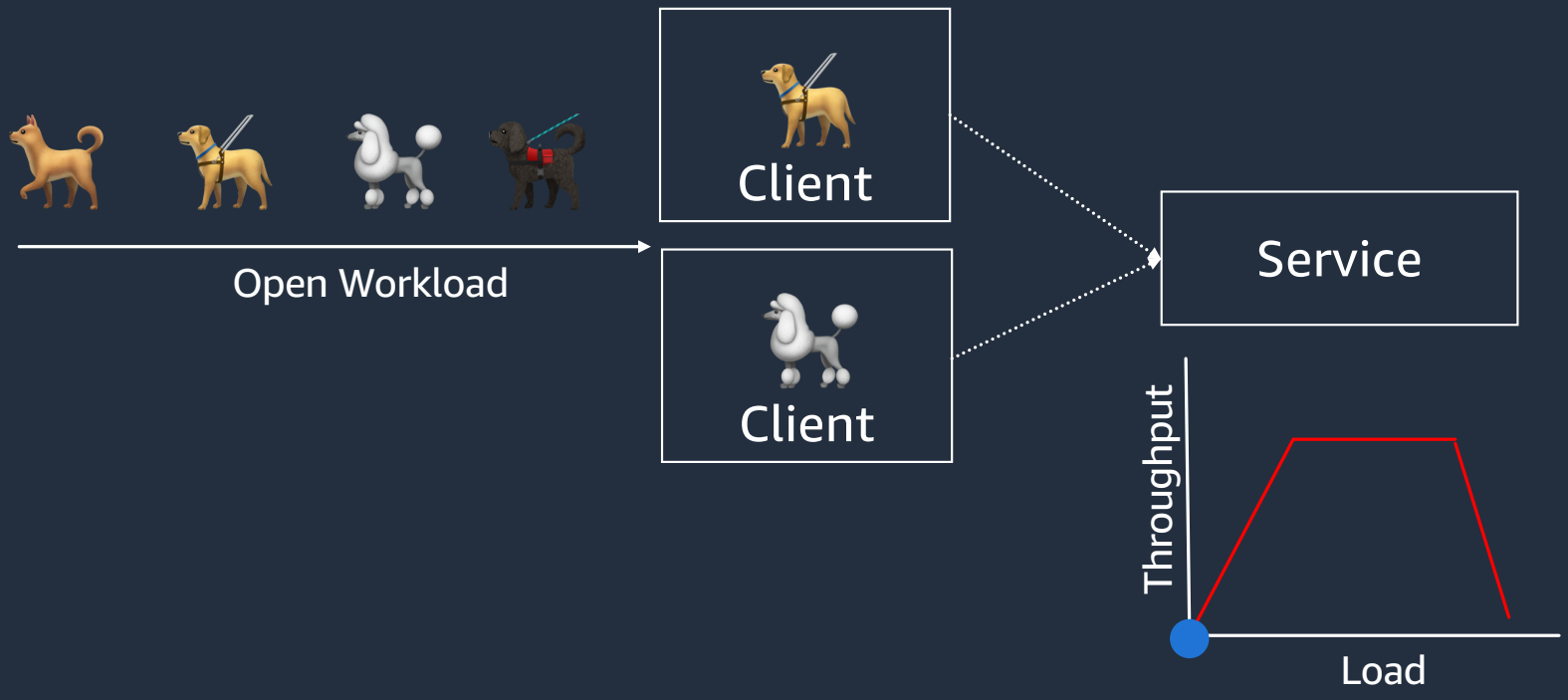
GOOD PERFORMANCE FOR BAD DAYS



GOOD PERFORMANCE FOR BAD DAYS







Research Roadmap 3:

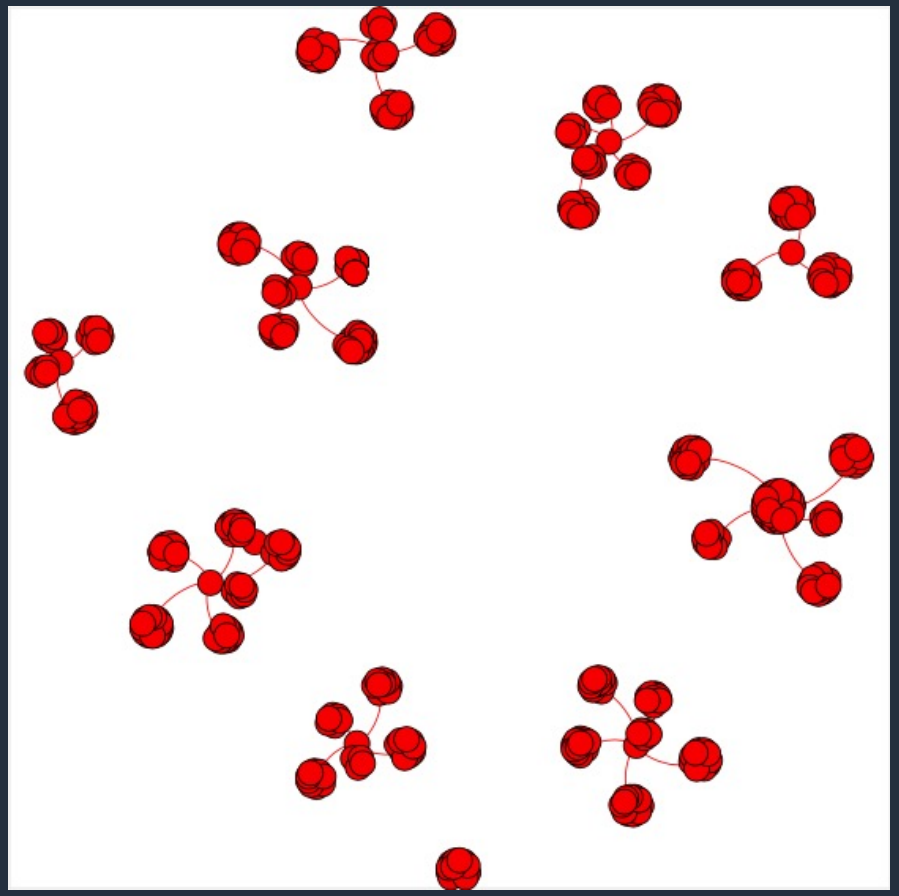
- Deeply understand metastable system behaviors.
- Formal and simulation tools to explore these behaviors *in silico.*
- Develop and advocate for better best practices.
- Algorithms for shedding load, detecting overload, etc.
- Incentives for client co-operation.

Why 4: Common Benchmarks Don’t Pay Attention to *Workloads*

In many systems (especially RDBMSs), performance is very sensitive to workload.

This is well known, but not well used.

GOOD PERFORMANCE FOR BAD DAYS



10 Warehouses
=
10 *Clusters*
=
Easy scaling

Real world workloads are not as kind.

Table 3: Query Runtimes: *Number of queries and their total runtimes grouped into various runtime buckets.*

Time bucket	% of queries			% of sum(runtime)		
	Fleet	TPC-H	-DS	Fleet	TPC-H	-DS
(0s, 10ms]	13.7	0	0	0.01	0	0
(10ms, 100ms]	48.3	0	0	0.4	0	0
(100ms, 1s]	24.9	0	22	2.3	0	2
(1s, 10s]	9.9	27	59	7.3	3	19
(10s, 1min]	2.2	55	13	13.3	30	25
(1min, 10min]	0.86	8	5	35.7	66	55
(10min, 1h]	8e-2	0	0	25.2	0	0
(1h, 10h]	8e-3	0	0	14.3	0	0
>=10h	9e-5	0	0	1.6	0	0

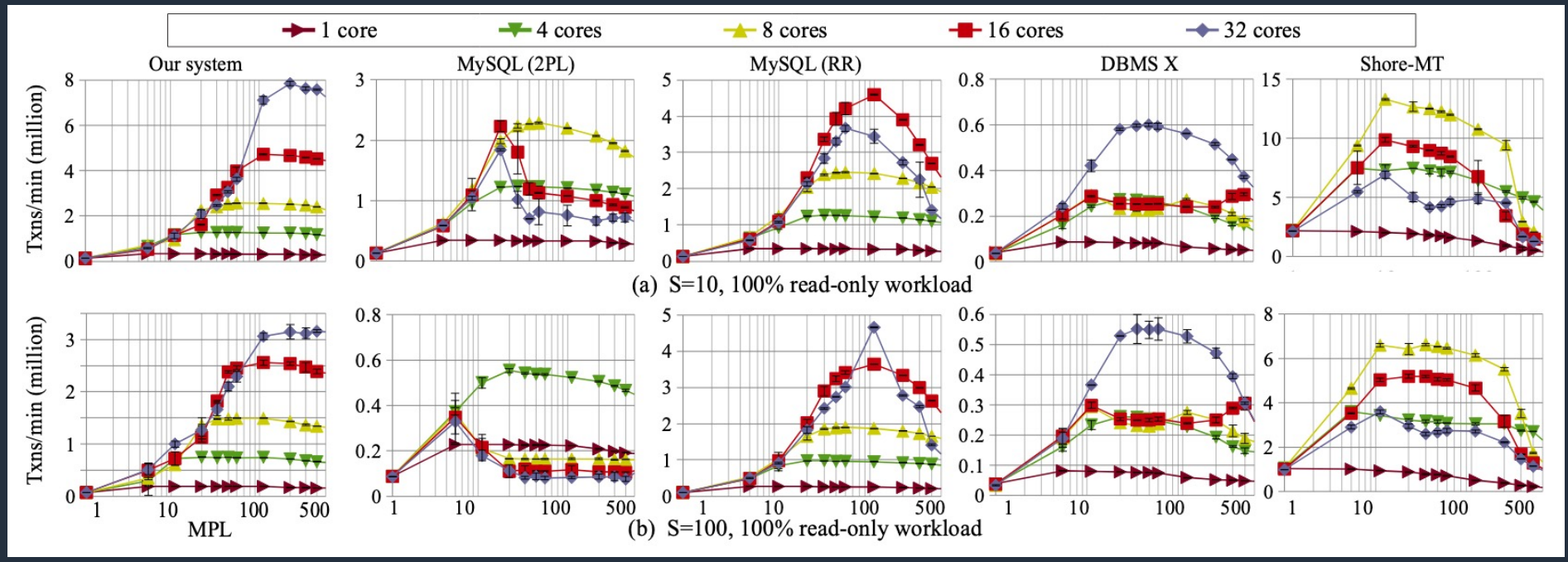
Van Renen, et al. “Why TPC Is Not Enough: An Analysis of the Amazon Redshift Fleet”, VLDB’24

Research Roadmap 4:

- Characterization at the graph level.
- Focus on need for co-ordination, at fundamental level.
- Workload generation, and space exploration.

Thank you!

Marc Brooker
mbrooker@amazon.com
https://brooker.co.za/blog/



Jung et al, *A Scalable Lock Manager for Multicores,* SIGMOD'13